SWDML501 MACHINE LEARNING

Learining Unit 1: ´Apply Data Preprocessing

1.1 Introduction to machine learning

1.1.1 Machine learning overview

Definition

Machine Learning (ML) is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that enable computer systems to perform a task without explicit programming. The primary goal of machine learning is to allow computers to **learn from data and impro**ve their performance over time.

In traditional programming, humans explicitly write code to instruct a computer on how to perform a specific task. In contrast, machine learning algorithms learn **patterns and relationships from data, enabling them to make predictions or decisions** without being explicitly programmed for each scenario.

Machine learning life cycle

The machine learning lifecycle encompasses the entire process of developing, deploying, and maintaining a machine learning model. The general overview of the machine learning lifecycle:

1. **Define the Problem**

Clearly articulate the problem you want to solve and determine if machine learning is the appropriate solution.

2. **Collect and Prepare Data**

Gather relevant data that will be used **to train, validate, and test the** machine learning model.

Clean and preprocess the data, handling missing values, outliers, and ensuring data is in a suitable format.

3. **Exploratory Data Analysis (EDA)**

Understand the characteristics of the data through visualization and statistical analysis.

Identify patterns, correlations, and potential features for the model.

4. **Feature Engineering**

Select or create features that are most relevant to the problem.

Transform and scale features as needed for model training.

5. **Select a Model**

Choose a machine learning algorithm or a combination of algorithms based on the nature of the problem and the characteristics of the data.

Split the Data:

Divide the dataset into training, validation, and test sets to assess the model's performance accurately.

6. **Train the Model**

Feed the training data into the selected model and adjust its parameters to learn patterns from the data.

Validate the model on the validation set to fine-tune hyperparameters and prevent overfitting.

7. **Evaluate the Model**

Assess the model's performance on the test set to ensure it generalizes well to new, unseen data.

Metrics such as accuracy, precision, recall, and F1 score are commonly used for evaluation.

8. **Deploy the Model**

If the model meets the desired performance criteria, deploy it to a production environment where it can make predictions on new data.

1.1.2 Applications

Machine learning has found applications in a wide range of fields, transforming industries and improving various processes.

Image and Speech Recognition

Machine learning is used in image recognition for tasks such as facial recognition, object detection, and image classification. Similarly, it is applied in speech recognition for converting spoken language into text.

Natural Language Processing (NLP)

NLP involves the interaction between computers and human language. Machine learning is applied in tasks such as sentiment analysis, language translation, chatbots, and text summarization.

Healthcare

Machine learning is employed for predictive analytics, disease identification, personalized treatment plans, drug discovery, and medical image analysis, improving diagnostics and patient care.

Finance:

In finance, machine learning is used for credit scoring, fraud detection, algorithmic trading, portfolio management, and risk assessment. These applications help optimize financial processes and enhance decision-making.

Recommendation Systems

E-commerce platforms, streaming services, and content providers use machine learning to create recommendation systems that suggest products, movies, or content based on user preferences and behavior.

Autonomous Vehicles

Machine learning plays a crucial role in developing self-driving cars. It enables vehicles to perceive their surroundings, make decisions, and navigate safely through various environments.

1.1.3 Advantages and disadvantages

1 Easily identifies trends and patterns

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves

to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2 No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3 Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in d

Disadvantages of Machine Learning

With all those advantages to its powerfulness and popularity, Machine Learning isn't perfect. The following factors serve to limit it

1. **Data Acquisition**

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. **Time and Resources**

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. **Interpretation of Results**

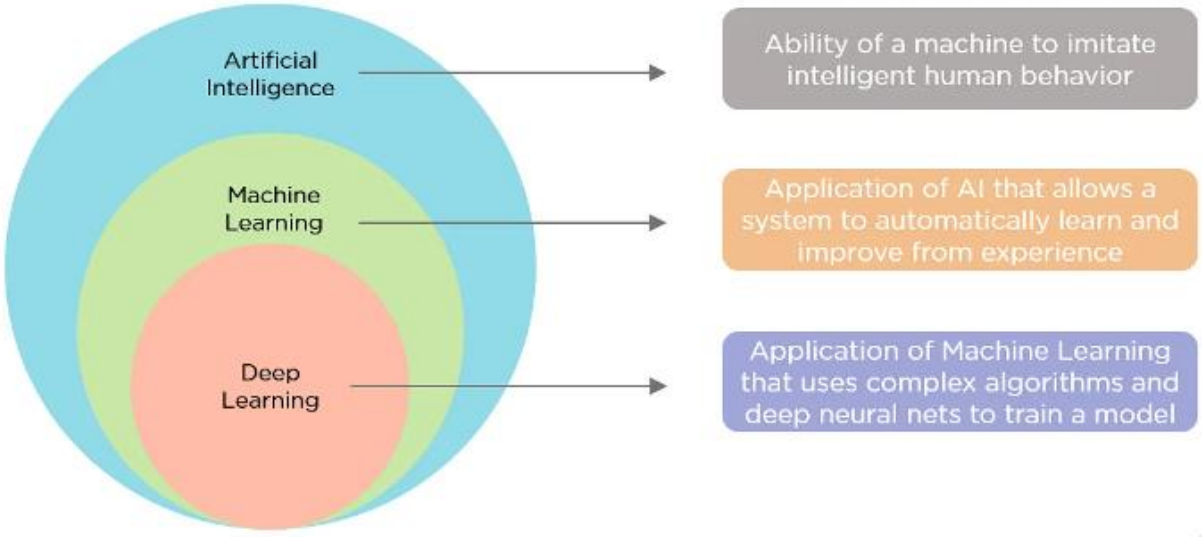
Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. **High error-susceptibility**

ML is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

1.1.4 Difference between machine learning, artificial intelligence, and deep learning

Here is an illustration designed to help us understand the fundamental differences between artificial intelligence, machine learning, and deep learning.



Artificial intelligence is the overarching system. Machine learning is a subset of AI. Deep learning is a subfield of machine learning, and neural networks make up the backbone of deep learning algorithms.

Artificial Intelligence, deep learning, machine learning—whatever you're doing if you don't understand it—learn it. Because otherwise, you're going to be a dinosaur within 3 years.

The three terms are often used interchangeably, but they do not quite refer to the same things.

Here is an illustration designed to help us understand the fundamental differences between artificial intelligence, machine learning, and deep learning.

Artificial Intelligent, the broadest term of the three, is used to classify machines that mimic human intelligence and human cognitive functions like problem-solving and learning. AI uses predictions and automation to optimize and solve complex tasks that humans have historically done, such as facial and speech recognition, decision making and translation.

Machine learning is a subset of artificial intelligence that allows for optimization. When set up correctly, it helps you make predictions that minimize the errors that arise from merely guessing. For example, companies like Amazon use machine learning to recommend products to a specific customer based on what they've looked at and bought before.

As our article on **deep learning** explains, deep learning is a subset of machine learning. The primary difference between machine learning and deep learning is how each algorithm learns and how much data each type of algorithm uses.

Neural networks, also called artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are the backbone of **deep learning algorithms**. They are called "neural" because they mimic how neurons in the brain signal one another.

1.1.2 Types of Machine Learning

- **Supervised Learning**

Supervised learning uses **labeled data** (data with known answers) to train algorithms to:

- Classify Data
- Predict Outcomes

Supervised learning can **classify** data like "What is spam in an e-mail", based on known spam examples.

Supervised learning can **predict** outcomes like predicting what kind of video you like, based on the videos you have played.

- **Unsupervised Learning**

In this type, the algorithm is given unlabeled data and must find patterns or relationships within the data on its own. Clustering and dimensionality reduction are common tasks in unsupervised learning.

Unsupervised learning is used to predict undefined relationships like meaningful patterns in data.

It is about creating computer algorithms than can improve themselves.

It is expected that machine learning will shift to unsupervised learning to allow programmers to solve problems without creating models.

- **Reinforcement Learning**

Involves training a model to make sequences of decisions by rewarding or penalizing the model based on the outcomes of its actions. The model learns to maximize cumulative rewards over time.

Reinforcement learning is based on non-supervised learning but receives feedback from the user whether the decisions is good or bad. The feedback contributes to improving the model.

1.1.3 Programming languages and tools

Programming languages involved in **Machine Learning** and Artificial Intelligence are:

- LISP
- R
- Python
- C++
- Java
- JavaScript
-

LISP

LISP is the second oldest programming language in the world (1958), one year younger than Fortran (1957).

The term **Artificial Intelligence** was made up by **John McCarthy** who invented LISP.

LISP was founded on the theory of **Recursive Functions** (a function appears in its own definition).

Recursive Functions can be written as self-modifying functions, and this is very suitable for Machine Learning programs where "self-learning" is an important part of the program

The R Language

R is a programming language for **Graphics** and **Statistical** computing.

R comes with a wide set of statistical and graphical techniques for:

- Linear Modeling
- Nonlinear Modeling
- Statistical Tests
- Time-series Analysis
- Classification
- Clustering

Python

Python is a general-purpose coding language. It can be used for all types of programming and software development.

Python is typically used for server development, like building web apps for web servers.

Python is also typically used in **Data Science**.

An advantage for using Python is that it comes with some very suitable libraries:

- NumPy (Library for working with Arrays)
- SciPy (Library for Statistical Science)
- Matplotlib (Graph Plotting Library)

- NLTK (Natural Language Toolkit)
- TensorFlow (Machine Learning)

C++

C++ holds the title: "**The worlds fastest programming language**".

Because of the speed, C++ is a preferred language when programming Computer Games.

It provides faster execution and has less response time which is applied in search engines and development of computer games.

Google uses C++ in Artificial Intelligence and Machine Learning programs for SEO (Search Engine Optimization).

SHARK is a super-fast C++ library with support for supervised learning algorithms, linear regression, neural networks, and clustering.

MLPACK is also a super-fast machine learning library for C++.

Java

Java is another general-purpose coding language that can be used for all types of software development.

For Machine Learning, Java is mostly used to create algorithms, and neural networks.

1.2 Setting up machine learning environment

The choice of tool depends on the specific requirements of the project and the preferences of the user or development team.

There are numerous machine learning tools and frameworks available, catering to different needs and preferences. Here are some popular ones:

Scikit-Learn: Scikit-Learn is an open-source machine learning library for Python. It provides simple and efficient tools for data analysis and modeling, including various algorithms for classification, regression, clustering, and more.

TensorFlow:Description: Developed by Google, TensorFlow is an open-source machine learning framework widely used for building and training deep learning models. It supports both neural networks and traditional machine learning algorithms.

PyTorch: PyTorch is an open-source deep learning framework developed by Facebook. It is known for its dynamic computation graph, making it flexible and suitable for research and experimentation in addition to production use.

Keras:Description: Keras is a high-level neural networks API that can run on top of **TensorFlow, Theano, or Microsoft Cognitive Toolkit**. It simplifies the process of building and experimenting with deep learning models.

Scikit-Image:Description: This is an image processing library built on top of Scikit-Learn. It provides algorithms for segmentation, feature extraction, and other image processing tasks.

Pandas:Description: While not a machine learning library per se, Pandas is a powerful Python library for data manipulation and analysis. It is often used for preparing and cleaning data before feeding it into machine learning models.

NLTK (Natural Language Toolkit): NLTK is a library for working with human language data. It provides easy-to-use interfaces to work with linguistic data for tasks such as classification, tokenization, stemming, tagging, parsing, and more.

RapidMiner: RapidMiner is an integrated data science platform that provides tools for data preparation, machine learning, and model deployment. It offers a visual interface for designing and executing machine learning workflows.

Microsoft Azure Machine Learning:Azure ML is a cloud-based service that provides tools for building, deploying, and managing machine learning models on the Microsoft Azure cloud. It supports various frameworks, including TensorFlow and PyTorch.

Google Colab:Google Colab is a free, cloud-based platform that provides a Jupyter notebook environment with free GPU and TPU support. It is often used for collaborative machine learning projects and experiments.

1.3 Data Collection and Acquisition

data collection is in fact the first and most fundamental step in the machine learning pipeline. It's part of the complex data processing phase within an ML lifecycle. From this comes another important point: data collection directly impacts the performance of an ML model and the final results.

Data acquisition is the process of taking measurements of real-world physical occurrences using signals and digitizing them so that a computer and software may alter them.

t is the procedure of locating relevant business data, formatting the information into the necessary business form, and loading the data into the specified system. he three fundamental parts of every data acquisition system are a sensor, signal conditioning, and an analog-to-digital converter (ADC).

1.3.2 Explanation of data, big data and ML dataset

With Machine Learning, **data** is collections of facts:

Type	Examples
Numbers	Prices. Dates.
Measurements	Size. Height. Weight.
Words	Names and Places.
Observations	Counting Cars.
Descriptions	It is cold.

Big Data

Big data is data that is impossible for humans to process without the assistance of advanced machines.

Big data does not have any definition in terms of size, but datasets are becoming larger and larger as we continously collect more and more data and store data at a lower and lower cost.

Data Set

In the mind of a computer, a data set is any collection of data. It can be anything from an array to a complete database.

Machine learning datasets are important for two reasons: they allow you **to train** your machine learning models, and they provide a benchmark **for measuring the accuracy of** your models.

1.3.3 Key Characteristics for ML dataset

high-quality machine learning dataset should possess accurate and reliable data (Quality), an adequate number of instances (Quantity), and sufficient diversity to cover various scenarios (Variability).

Quality

Quality refers to the reliability and accuracy of the data within the dataset. High-quality data is free from errors, inconsistencies, and inaccuracies.

High-quality data contributes to **accurate and reliable predictions** by ensuring that the model learns meaningful patterns from the data.

Reliable data enhances the **trustworthiness** of the machine learning model, making it more suitable for decision-making in real-world scenarios.

Quality data **reduces the likelihood of biases in the** dataset, which can lead to biased model predictions.

Quantity

Quantity refers to the **size of the dataset**, measured in terms of the number of instances or data points. A larger dataset generally provides more information for training a robust model.

Larger datasets often lead to **better generalization**, allowing the model to perform well on **new, unseen data.**

Adequate data helps **prevent overfitting, where** the model **memorizes** the training data instead of learning general patterns.

A larger dataset provides more statistically **significant insights and patterns,** leading to more reliable conclusions.

Variability

Variability refers to the diversity and range of the data within the dataset. A dataset with sufficient variability covers different scenarios, conditions, and potential outcomes.

Importance:

A dataset with variability helps **train a robust** model that can **handle diverse situations** and variations present in real-world data.

Avoiding Overfitting: Including diverse examples reduces the risk of overfitting to specific patterns present in the training data but not applicable to new instances.

Variability contributes to better generalization, allowing the model to make accurate predictions across different contexts

1.3.4 Types of datasets

A machine learning dataset is a collection of data that is used to train the model. A dataset acts as an example to teach the machine learning algorithm how to make predictions. The common types of data include.

- Text data
- Image data
- Audio data
- Video data
- Numeric data

Source of Dataset

Sourcing a dataset depends on the requirements and scope of the project. If your project does not require highly personalized data, some common datasets that can be sourced from vendors for different types of projects.

1. Natural language processing datasets

Datasets are used for, text analytics, and language translation. These types of datasets are large in size and require heavy computational power.

Some popular NLP datasets include:

Clickworker datasets

Amazon Reviews

The Big ad NLP Database

Wikipedia Links Data

2. **Open datasets**

These ready-to-use datasets are freely available online for anyone to download, modify, and distribute without legal or financial restrictions. These datasets are regularly updated and are compatible with most ML frameworks. The only drawback is that open datasets lack personalization.

Popular open datasets include:

Google dataset search

AWS public datasets

Kaggle datasets

3. **Public government datasets**

These datasets are used for government projects that are implemented for the public. For example, these datasets can include a certain population's census or demographic data. These datasets can be used to make policies or train AI/ML models for immigration decision-making, chatbots that answer citizen queries, city infrastructure planning systems, etc.

Popular public government datasets include:

- Data.Gov.uk
- EU open data portal
- Data USA

4. **Image datasets**

Image datasets include both image and video data. This type of dataset is used to train computer vision systems for facial recognition, autonomous vehicle systems, retail security systems, etc. These datasets required high-quality image annotation to be used.

Popular image datasets include:

- Google's open images
- Coco dataset
- Imagenet
- Baidu ApolloScape Dataset
- Waymo Open Dataset

5. **Audio datasets**

These datasets are used to train AI/ML models for voice recognition, music recognition, etc.

Popular audio datasets include:

- Environmental audio datasets
- Speech commands dataset
- Free music archive (FMA)
- Flickr audio caption corpus

Common voice

6. **Healthcare Datasets**

These datasets are used to train medical imaging systems or medical diagnosis systems. They are usually large in size and require heavy computational and high-quality medical annotation.

Popular healthcare datasets include:

MIMIC Critical Care Database

Healthdata.gov

You can also check out our data-driven list of data collection/harvesting services to find the option that best suits your project needs.

7. **Generative AI**

Generative AI, particularly for models like Generative Adversarial Networks (GANs), has transformed the landscape of data creation and augmentation. Creating datasets using generative AI addresses several challenges in machine learning.

n situations where collecting real-world data is expensive, time-consuming, or ethically challenging, generative models can supplement or even replace traditional data collection methods.

For instance, medical imaging datasets can be augmented using GANs to generate more samples of rare conditions, making it easier to train models that can detect and diagnose these conditions.

Additionally, in domains like computer vision (CV), generating diverse data helps in mitigating overfitting and improving the robustness of the trained models. This synthesized data, when used judiciously alongside real data, can help to train more effective and accurate machine learning models, while saving resources and time in the data collection phase.

1.4 Data Preprocessing

Data Processing is the task of converting data from a given form to a much more usable and desired form i.e. making it more meaningful and informative. Using Machine Learning algorithms, mathematical modeling, and statistical knowledge, this entire process can be automated. The output of this complete process can be in any desired form like graphs, videos, charts, tables, images, and many more, depending on the task we are performing and the requirements of the machine.

The **goal** of data processing is to clean, transform, and prepare the data in a format that is suitable for modeling.

1. **The main steps involved in data processing typically include:**

- **Data collection**: This is the process of gathering data from various sources, such as sensors, databases, or other systems. The data may be structured or unstructured, and may come in various formats such as text, images, or audio.
- **Data preprocessing:** This step involves cleaning, filtering, and transforming the data to make it suitable for further analysis. This may include removing **missing values, scaling or normalizing the data, or converting it to a different format.**
- **Data analysis**: In this step, the data is analyzed using various techniques such as statistical analysis, machine learning algorithms, or data visualization. The goal of this step is to derive insights or knowledge from the data.
- **Data interpretation**: This step involves interpreting the results of the data analysis and drawing conclusions based on the insights gained. It may also involve presenting the findings in a clear and concise manner, such as through reports, dashboards, or other visualizations.
- **Data storage and management**: Once the data has been processed and analyzed, it must be stored and managed in a way that is secure and easily accessible. This may involve storing the data in a database, cloud storage, or other systems, and implementing backup and recovery strategies to protect against data loss.
- **Data visualization and reporting: Finally, the results of the data analysis are presented to stakeholders in a** format that is easily understandable and actionable. This may involve creating visualizations, reports, or dashboards that highlight key findings and trends in the data.

2. Characteristics of quality Data

Accuracy refers to the correctness and precision of the data. Accurate data is free from errors and reflects the true values or characteristics it is intended to represent.

Completeness measures the extent to which the dataset includes all the necessary information. A complete dataset contains all the expected records and fields without missing values.

Consistency refers to the uniformity and standardization of data across the dataset. Consistent data maintains the same format, units, and conventions throughout.

Relevance assesses the significance of the data to the goals and objectives of the analysis or modeling. Relevant data is directly related to the problem at hand.

Validity refers to the extent to which the data accurately represents the real-world entities or phenomena it is supposed to capture. Valid data is meaningful and appropriate for its intended use.

3. Data cleaning for inconsistencies rectification

Data cleaning is one of the important parts of machine learning. It plays a significant part in building a model. However, the success or failure of a project relies on proper data cleaning. Professional data scientists usually invest a very large portion of their time in this step because of the belief that "Better data beats fancier algorithms".

Data cleaning, also known as data cleansing or data preprocessing, is a crucial step in the data science pipeline that involves identifying and correcting or removing errors, inconsistencies, and inaccuracies in the data to improve its quality and usability. Data cleaning is essential because raw data is often noisy, incomplete, and inconsistent, which can negatively impact the accuracy and reliability of the insights derived from it.

- Import the necessary libraries
- Load the dataset
- Check the data information using df.info()

```
import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv('train.csv')
df.head()
```

1. Data inspection and exploration:

This step involves understanding the data by inspecting its structure and identifying missing values, outliers, and inconsistencies.

- Check the duplicate rows.

```
df.duplicated()
```

- Check the data information using df.info()

```
df.info()
```

2. Removal of unwanted observations

This includes deleting duplicate/ redundant or irrelevant values from your dataset. Duplicate observations most frequently arise during data collection and Irrelevant observations are those that don't actually fit the specific problem that you're trying to solve.

- Redundant observations alter the efficiency to a great extent as the data repeats and may add towards the correct side or towards the incorrect side, thereby producing unfaithful results.
- Irrelevant observations are any type of data that is of no use to us and can be removed directly.

Now we have to make a decision according to the subject of analysis, which factor is important for our discussion. As we know our machines don't understand the text data. So, we have to either drop or convert the categorical column values into numerical types. Here we are dropping the Name columns because the Name will be always unique and it hasn't a great influence on target variables.

```
df['Ticket'].unique()[:50]
```

It will the case of **Feature Engineering**. where we derived new features from a column or a group of columns. In the current case, we are dropping the "Name" and "Ticket" columns.

```
df1 = df.drop(columns=['Name','Ticket'])
df1.shape
```

4. Handling missing data:

Missing data is a common issue in real-world datasets, and it can occur due to various reasons such as human errors, system failures, or data collection issues. Various techniques can be used to handle missing data, such as imputation, deletion, or substitution.

Let's check the % missing values columns-wise for each row using df.isnull() it checks whether the values are null or not and gives returns boolean values. and .sum() will sum the total number of null values rows and we divide it by the total number of rows present in the dataset then we multiply to get values in % i.e per 100 values how much values are null.

```
round((df1.isnull().sum()/df1.shape[0])*100,2)
```

So, we will drop the Cabin column. Embarked column has only 0.22% of null values so, we drop the null values rows of Embarked column.

```
df2 = df1.drop(columns='Cabin')
df2.dropna(subset=['Embarked'], axis=0, inplace=True)
df2.shape
```

- Imputing the missing values from past observations.
    - Again, "missingness" is almost always informative in itself, and you should tell your algorithm if a value was missing.
    - Even if you build a model to impute your values, you're not adding any real information. You're just reinforcing the patterns already provided by other features.

From the above describe table, we can see that there are very less differences between the mean and median i..e 29.6 and 28. So, here we can do any one from mean imputation or Median imputations.

Note:

- Mean imputation is suitable when the data is normally distributed and has no extreme outliers.
- Median imputation is preferable when the data contains outliers or is skewed.

```
# Mean imputation
df3 = df2.fillna(df2.Age.mean())
# Let's check the null values again
df3.isnull().sum()
```

5. **Handling outliers:**

Outliers are extreme values that deviate significantly from the majority of the data. They can negatively impact the analysis and model performance. Techniques such as clustering, interpolation, or transformation can be used to handle outliers.

```
import matplotlib.pyplot as plt

plt.boxplot(df3['Age'], vert=False)
plt.ylabel('Variable')
plt.xlabel('Age')
plt.title('Box Plot')
plt.show()
```

6. **Data transformation**

Data transformation involves converting the data from one form to another to make it more suitable for analysis. Techniques such as normalization, scaling, or encoding can be used to transform the data.

- **Data validation and verification:** Data validation and verification involve ensuring that the data is accurate and consistent by comparing it with external sources or expert knowledge.

For the machine learning prediction, First, we separate independent and target features. Here we will consider only **'Sex' 'Age' 'SibSp', 'Parch' 'Fare' 'Embarked'** only as the independent features and **Survived** as target variables. Because PassengerId will not affect the survival rate.

```
X = df3[['Pclass','Sex','Age', 'SibSp','Parch','Fare','Embarked']]
Y = df3['Survived']
```

Data Normalization

Significance of Data Normalization: In the realm of data analysis and machine learning, data normalization plays a pivotal role by ensuring uniform scaling across all features. This is particularly critical for models relying on metrics like distance calculations (e.g., k-nearest neighbors) or optimization based on gradients (e.g., neural networks). The normalization process

prevents certain features from exerting undue influence on others, leading to a more resilient and precise model.

Data Normalization Technoiques:

Min-Max Scaling:Adjusts data to a specified range (e.g., [0, 1]) by subtracting the minimum value and dividing by the range.

Z-Score Normalization (Standardization):Transforms data to exhibit a mean of 0 and a standard deviation of 1.

Robust Scaling:Scales data based on the median and interquartile range, rendering it less susceptible to outliers.

Decimal Scaling:Alters data by relocating the decimal point of values, ensuring the largest absolute value becomes less than 1.

Data Transformation

Significance of Data Transformation: Data transformation is a vital process involving the conversion of data into a suitable format for analysis. It addresses issues such as skewness, heteroscedasticity, and non-linearity in the data, thereby enhancing model performance and interpretability.

Data Transformation Techniques:

Log Transformation:

Mitigates the impact of skewed data by taking the logarithm of the values.

Box-Cox Transformation:

Represents a generalized power transformation that stabilizes variance and imparts a more normal distribution to the data.

Square Root Transformation:

Valuable for reducing the impact of right-skewed data.

Types of Data Transformation:

Linear Transformation:

Preserves the linear relationship between variables.

Non-Linear Transformation:

Introduces non-linearity to capture intricate patterns in the data.

Feature Engineering:

Significance of Feature Engineering: Feature engineering entails the creation or modification of features to enhance model performance, enabling models to comprehend underlying data patterns and improve predictive accuracy.

Steps:

Understanding the Data: Gaining insights into the dataset to identify potential features.

Feature Creation: Generating new features based on domain knowledge or data relationships.

Feature Selection: Choosing the most pertinent features to reduce dimensionality and enhance model efficiency.

Techniques and Tools:

One-Hot Encoding: Converts categorical variables into binary vectors.

Polynomial Features: Introduces polynomial terms to capture non-linear relationships.

Feature Scaling: Ensures consistency in the scale of all features.

PCA (Principal Component Analysis): Reduces dimensionality while preserving as much variance as possible.